Parallel and distributed optimization methods for estimation and control in networks ☆

Ion Necoara[a], Valentin Nedelcu[a], Ioan Dumitrache[a]

[a] *University Politehnica Bucharest, Automatic Control and Systems Engineering Department, 060042 Bucharest, Romania*
(e-mail: ion.necoara,ioan.dumitrache,v.nedelcu@upb.ro)

Abstract

System performance for networks composed of interconnected subsystems can be increased if the traditionally separated subsystems are jointly optimized. Recently, parallel and distributed optimization methods have emerged as a powerful tool for solving estimation and control problems in large-scale networked systems. In this paper we review and analyze the optimization-theoretic concepts of parallel and distributed methods for solving coupled optimization problems and demonstrate how several estimation and control problems related to complex networked systems can be formulated in these settings. The paper presents a systematic framework for exploiting the potential of the decomposition structures as a way to obtain different parallel algorithms, each with a different tradeoff among convergence speed, message passing amount and distributed computation architecture. Several specific applications from estimation and process control are included to demonstrate the power of the approach.

Keywords: Estimation, cooperative and distributed control, networks of interconnected subsystems, convex optimization, parallel and distributed methods, duality theory, consensus.

☆The research leading to these results has received funding from: the European Union, Seventh Framework Programme (FP7/2007–2013) under grant agreement no 248940; CNCSIS-UEFISCSU (project TE, no. 19/11.08.2010); ANCS (project PN II, no. 80EU/2010); Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

1. Introduction

In many application fields, the notion of networks has emerged as a central, unifying concept for solving different problems in systems and control theory such as analysis, process control and estimation. We live and operate in a networked world. We drive to work on networks of roads and communicate with each other using an elaborate set of devices such as phones or computers, that connect wirelessly and through the internet. Traditional networks include transportation networks (roads, rails) and networks of utilities (water, electricity, gas). But more recent examples of the increasing impact of networks include information technology networks (internet, mobile phones, acoustic networks, etc), information networks (co-author networks, bibliographic networks), social networks (collaborations, organizations), and biological and genetic networks.

These networks are often composed of multiple subsystems characterized by complex dynamics and mutual interactions such that local decisions have long-range effects throughout the entire network. Many problems associated to networked systems, such as state estimation and control, can be posed as *coupled optimization problems* (see e.g. [4, 10, 11, 17, 21, 25, 28, 49], etc). Note that in these systems the interaction between subsystems gives rise to coupling in the cost or constraints, but with a specific algebraic structure, in particular sparse matrix representation that could be exploited in numerical algorithms. Therefore, in order to design an overall decision architecture for such complex networks we need to solve large coupled optimization problems but with specific structure. The major difficulty in these problems is that due to their size, communication restrictions, or requirements on robustness, often no central decisions can be taken; instead, the decisions have to be taken locally. In such a set-up, single units, or local agents, must solve local optimization subproblems and then they must negotiate their outcomes and requirements with their neighbors in order to achieve convergence to the global optimal solution. Basically, there are two general optimization approaches:

(i) "Centralized" optimization algorithms: In this class the specific structure of

the system is exploited, as it represents considerable sparsity in the optimization problem due to the local coupling between optimization variables (sometimes referred to as *separable optimization problems*). The sparsity of the problem, given by the influences between the subsystems, leads to coupling constraints represented by sparse matrices. Though parts of the algorithms will be parallelized, the parallelization in these algorithms is not restricted by e.g. limited communication between subsystems and is just for the sake of exploiting sparsity. In summary, "centralized" algorithms benefit from the sparsity induced by the networked system and solve the resulting optimization problems on a parallel computer architecture. Several standard parallel and distributed optimization methods can be found in the textbooks [2, 18, 22]. Various survey papers also exist on optimization-based distributed control. In the 70's Tamura [47] and Mahmoud [23] presented very comprehensive overviews. More recently, in [42] the actual status of research in the field of coordinated optimization-based control is presented. Many different control topologies can be considered in distributed control, which have been reviewed recently in [44]. When there is no need to solve the separable optimization problem on a parallel computer architecture, an alternative would be to solve the global optimization problem using sparse solvers that take into account the sparse structure of the problem at the linear algebra level of the optimization algorithm. In general, this choice could lead to faster algorithms in terms of CPU time than distributed or parallel algorithms.

(ii) Distributed optimization algorithms (sometimes referred to as *distributed multi-agent optimization algorithms*): In contrast to the "centralized" algorithms, distributed algorithms on graphs have to satisfy an extra constraint, namely their computations shall be performed on all nodes in parallel, and the communication between nodes is restricted to the edges of the graph, i.e. such algorithms do not use all-to-all communication protocols. In many complex networked systems the desired behavior can be formulated as coupled optimization problems but with restrictions on communication due to the special network topology: e.g. estimation in sensor networks, consensus and ren-

dezvous problems in multi-agent systems, resource allocation in computer networks [10, 13, 35]. Some existing distributed methods that take into account explicitly information restrictions in the network combine consensus negotiations (as an efficient method for information fusion) with subgradient methods [14, 26, 31, 32, 52].

The goal of this paper is twofold: (i) to establish a relationship between estimation and control in networked systems and distributed optimization methods and demonstrate the effectiveness of utilizing optimization-theoretic approaches for controlling such complex systems; (ii) motivated by this connection, to build upon optimization based results to better accommodate a broader class of estimation and control problems. The core of this paper consists of Section 2, covering three applications of estimation and control that appear in the context of networked systems and then proving how we can reformulate them as coupled optimization problems. One of the key contributions of this paper is to provide an accessible, yet relatively comprehensive, overview of three classes of decomposition schemes from mathematical programming for solving distributively coupled optimization problems. We demonstrate how the decomposition schemes suggest network architectures and protocols with different properties in terms of convergence speed and coordination overhead. We also present new decomposition methods that are more efficient in terms of convergence speed than some classical decomposition schemes.

The paper is organized as follows. In Section 2 we introduce different estimation and control problems that appear in the context of complex systems with interacting subsystems dynamics and then we show how we can reformulate them as coupled optimization problems. In Section 3 we present several parallel and distributed methods for solving this type of structured optimization problems and analyze their performance. Section 3 thus serves both as a review of the necessary background and a summary of our new extensions on decomposition methods. For each of the applications, numerical experiments on different parallel and distributed algorithms are provided.

2. Estimation and control problems in networks

In this section we formulate different estimation and control problems for systems consisting of interconnected subsystems. In Subsection 2.1 we present a state estimation problem for a system, using a network of sensors which must exchange information in order to reach a consensus on the state estimated for the entire system. In Subsection 2.2 we will present the problem of optimal control for a large-scale system, whose subsystems are coupled with their neighbors but the objective function is decoupled. Finally, in Subsection 2.3 and 2.4 we will discuss the cooperative control problem for a group of systems (agents), which have decoupled or coupled dynamics but share a common goal.

2.1. State estimation problem

In this section we formulate the distributed state estimation problem for systems using a sensor network based on the moving horizon estimation (MHE) approach [10, 11, 12, 40, 41]. Sensor networks can be employed in many applications, such as monitoring, exploration, surveillance or tracking targets over specific regions. We consider the concept of MHE, as this framework offers multiple advantages: since a particular minimization problem must be solved on-line at each step, the observer is optimal with respect to the associated cost, and moreover, constraints on the state and on the noise can be taken into account [11, 12, 40, 41].

The state estimation problem can be posed as follows. We assume that each sensor in the network measures some variables of a process, computes a local estimate of the entire state of the system, and exchanges the computed estimates with its neighbors. The solution to the estimation problem consists in finding a methodology which guarantees that all sensors asymptotically reach a reliable estimate of the overall state of the system. For the observed process we consider the following nonlinear dynamics:

$$x_{t+1} = \phi(x_t) + w_t,$$

where $x_t \in X \subseteq \mathbb{R}^n$ is the state vector and $w_t \in W \subseteq \mathbb{R}^n$ represents a white noise with covariance equal to Q. We also assume that the sets X and W are convex. The initial condition x_0 is a random variable with mean $\hat{x}_0$ and covariance Π_0. Measurements on the state vector are performed by M sensors[1], according to the following sensing model:

$$y_t^i = \theta^i(x_t) + v_t^i, \quad \forall i = 1, \cdots, M$$

where $v_t^i \in \mathbb{R}^{p_i}$ represents white noise with covariance matrix R_i. The functions ϕ and θ^i can be in general nonlinear.

For a given estimation horizon $N \geq 1$, at time k given the past measurements $y_{k-N}^i, \cdots, y_k^i$ provided by the ith sensor and the estimate $\hat{x}_{k-N}$, we formulate the moving horizon estimation (MHE) at k as the solution to the following optimization problem [12, 40, 41]:

$$\min_{x_{k-N}, w_t} \sum_{i=1}^{M} \sum_{t=k-N}^{k} ||v_t^i||^2_{R_i^{-1}} + \sum_{t=k-N}^{k-1} ||w_t||^2_{Q^{-1}} + ||x_{k-N} - \hat{x}_{k-N}||^2_{\Pi_{k-N}^{-1}} \tag{1}$$

$$\text{s.t.}: \; x_{t+1} = \phi(x_t) + w_t, \tag{1.1}$$

$$x_t \in X, \; w_t \in W \quad \forall t, \tag{1.2}$$

where the matrix Π_{k-N} is computed recursively from a Riccati difference equation in a centralized way [40]. For the liner case, the distributed computation of this matrix can be done in many ways: e.g. using the steady-state MHE formulation (i.e. computing off-line Π_∞, which is the solution of the corresponding algebraic Riccati equation) or updating Π_{k-N} for all the sensors in the same way (using a common covariance matrix R for all sensors in the Riccati difference equation update). For the nonlinear case, the update of Π_{k-N} in a distributed fashion is still an open issue.

Note that $v_t^i = y_t^i - \theta^i(x_t)$ and using the dynamics (1.1), we can write $\sum_{t=k-N}^{k} ||v_t^i||^2_{R_i^{-1}}$ as a function depending only on $(x_{k-N}, w_{k-N}, \cdots, w_{k-1})$.

[1] Throughout the paper we will use the convention that every superscript indicates a sensor/subsystem index.

Therefore, by eliminating the states in (1) using the dynamics (1.1) and introducing the notations:

$$\mathbf{x} = [x_{k-N}^T \; w_{k-N}^T \cdots w_{k-1}^T]^T,$$
$$f^i(\mathbf{x}) = \sum_{t=k-N}^{k} ||v_t^i||^2_{R_i^{-1}} + \frac{1}{M} \sum_{t=k-N}^{k-1} ||w_t||^2_{Q^{-1}} + \frac{1}{M}||x_{k-N} - \hat{x}_{k-N}||^2_{\Pi_{k-N}^{-1}},$$

the MHE problem (1) can be recast as an optimization problem with decoupled cost but a common decision variable $\mathbf{x}$ (**DCx**):

$$(\mathbf{DCx}) : \begin{cases} \min_{\mathbf{x}} \sum_{i=1}^{M} f^i(\mathbf{x}) \\ \text{s.t.}: \quad \mathbf{x} \in \mathbf{X}, \end{cases}$$

where the set $\mathbf{X} = X \times W^N$.

We assume that the communication network among sensors is described by a graph $G = (V, E)$, where the nodes in $V = \{1, \cdots, M\}$ represent the sensors and the edge $(i, j) \in E \subseteq V \times V$ models that sensor j sends information to sensor i. Then, the main challenge is to provide distributed algorithms for solving problem (1) or equivalently (**DCx**) which guarantee that all the sensors asymptotically reach a reliable estimate of the state variables using the information exchange model given by the graph G.

Example 2.1 In the particular case where the state and noise constraints $x \in X$ and $w \in W$ are described by linear inequalities (i.e. X and W are polyhedral sets) and the dynamics of the process and of the sensors are linear, i.e.

$$x_{t+1} = Ax_t + w_t,$$
$$y_t^i = C_i x_t + v_t^i, \quad \forall i = 1, \cdots, M,$$

the MHE problem (1) can be recast as a *separable convex quadratic program* with decoupled cost but a common decision variable in the form (**DCx**):

$$\min_{\mathbf{x}} \sum_{i=1}^{M} \mathbf{x}^T H_i \mathbf{x} + q_i^T \mathbf{x} \tag{2}$$
$$\text{s.t.}: \; \mathbf{x} \in \mathbf{X},$$

where the matrices H_i are positive definite and the constraint set $\mathbf{X}$ becomes in this case polyhedral (described only by linear inequalities).

2.2. Distributed optimal control problem

The application that we will discuss in this section is the distributed control of large-scale networked systems with interacting subsystem dynamics, which can be found in a broad spectrum of applications ranging from traffic networks, wind farms, to interconnected chemical plants. Distributed control is promising in applications for complex systems, since this framework allows us to design local subsystem-based controllers that take into account the interactions between different subsystems and physical constraints.

We consider discrete-time systems which can be decomposed into M subsystems described by difference equations of the form:

$$x_{t+1}^i = \phi^i(x_t^j, u_t^j; j \in \mathcal{N}^i), \quad \forall i = 1, \cdots, M, \tag{3}$$

where $x_t^i \in \mathbb{R}^{n_i}$ and $u_t^i \in \mathbb{R}^{m_i}$ represent the state and the input of the ith subsystem. The index set $\mathcal{N}^i$ contains the index i and all the indices of the subsystems which interact with the subsystem i. We also assume that the input and state sequences must satisfy local constraints:

$$x_t^i \in X^i, \quad u_t^i \in U^i, \quad \forall i = 1, \cdots, M, \quad \forall t \geq 0, \tag{4}$$

where the constraint sets $X^i \subseteq \mathbb{R}^{n_i}$ and $U^i \subseteq \mathbb{R}^{m_i}$ are usually compact sets. The system performance over a prediction horizon of length N is expressed through a stage cost and a final cost, which are composed of individual costs for each subsystem i and have the form:

$$\sum_{t=0}^{N-1} \ell^i(x_t^i, u_t^i) + \ell_f^i(x_N^i).$$

The centralized optimal control problem over a prediction horizon N reads:

$$\min_{x_t^i, u_t^i} \sum_{i=1}^{M} \sum_{t=0}^{N-1} \ell^i(x_t^i, u_t^i) + \sum_{i=1}^{M} \ell_f^i(x_N^i) \tag{5}$$

$$\text{s.t.}: \ x_0^i = x^i, \quad x_{t+1}^i = \phi^i(x_t^j, u_t^j; j \in \mathcal{N}^i), \tag{5.1}$$

$$x_t^i \in X^i, \ u_t^i \in U^i, \quad \forall t, i, \tag{5.2}$$

where x^i are the values of the initial state for subsystem i. Note that a similar formulation of distributed control for coupled subsystems with decoupled costs has been given in [4, 8, 25, 27, 28] in the context of distributed model predictive control.

Now, we show that the optimization problem (5) can be recast as a *separable optimization problem* with a particular structure. To this purpose, we denote with $\mathbf{X}^i = (X^i)^N \times (U^i)^N$ and

$$\mathbf{x}^i = [x_1^{iT} \cdots x_N^{iT} \ u_0^{iT} \cdots u_{N-1}^{iT}]^T,$$
$$f^i(\mathbf{x}^i) = \sum_{t=0}^{N-1} \ell^i(x_t^i, u_t^i) + \ell_f^i(x_N^i).$$

With these notations, problem (5) now reads as an optimization problem with decoupled cost and sparse coupled constraints (**DCCC**):

$$\textbf{(DCCC)}: \begin{cases} \min_{\mathbf{x}^1, \cdots, \mathbf{x}^M} \sum_{i=1}^{M} f^i(\mathbf{x}^i) \\ \quad \text{s.t.}: \ \mathbf{x}^i \in \mathbf{X}^i, \ h^i(\mathbf{x}^j; \ j \in \mathcal{N}^i) = 0 \ \ \forall i, \end{cases}$$

where the coupled constraints $h^i(\mathbf{x}^j; \ j \in \mathcal{N}^i) = 0$ are obtained from the coupling between the subsystems, i.e. by stacking the constraints (5.1) for a given i.

The centralized optimization problem (5) or (**DCCC**) becomes interesting if the computations can be distributed among the subsystems (agents), can be done in parallel and the amount of information that the agents must exchange is limited. In comparison with the centralized approach, a distributed strategy offers a series of advantages: first, the numerical effort is considerably smaller since we solve low dimension problems in parallel and secondly such a design is modular, i.e. adding or removing subsystems does not require any controller redesign.

Example 2.2 Many networked systems, e.g. wind farms [20], interconnected chemical processes [28, 50], or urban traffic systems [36], can be decomposed into M appropriate linear subsystems:

$$x_{t+1}^i = A_i x_t^i + B_i u_t^i + \sum_{j \in \mathcal{N}^{-i}} A_{ij} x_t^j + B_{ij} u_t^j, \quad \forall i = 1, \cdots, M, \tag{6}$$

where the index set $\mathcal{N}^{-i} = \mathcal{N}^i - \{i\}$, i.e. it contains all the indices of the subsystems which interact with the ith subsystem. If we introduce an auxiliary variable $w_t^i \in \mathbb{R}^{p_i}$ to represent the influence of the neighboring subsystems on the ith subsystem (in applications we usually have $p_i << n_i$), we can rewrite the dynamics (6) as:

$$x_{t+1}^i = A_i x_t^i + B_i u_t^i + E_i w_t^i, \quad \forall i,$$

where the matrices E_i are of appropriate dimensions and

$$w_t^i = \sum_{j \in \mathcal{N}^{-i}} A_{ij}^- x_t^j + B_{ij}^- u_t^j,$$

with the matrices A_{ij}^-, B_{ij}^- being obtained from the matrices A_{ij}, B_{ij} by removing the rows with all entries equal to zero. We consider a quadratic performance index for each subsystem i of the form:

$$\sum_{t=0}^{N-1} \left(||x_t^i||_{Q_i}^2 + ||u_t^i||_{R_i}^2 \right) + ||x_N^i||_{P_i}^2,$$

where the matrices Q_i, R_i and P_i are positive semidefinite. We also assume that the sets X^i and U^i that define the state and input constraints (4) are polyhedral. The centralized control problem over the prediction horizon N for this application can be formulated as follows:

$$\min_{x_t^i, u_t^i, w_t^i} \sum_{i=1}^{M} \sum_{t=0}^{N-1} ||x_t^i||_{Q_i}^2 + ||u_t^i||_{R_i}^2 + ||x_N^i||_{P_i}^2 \tag{7}$$

$$\text{s.t.} : \; x_0^i = x^i, \;\; x_{t+1}^i = A_i x_t^i + B_i u_t^i + E_i w_t^i, \tag{7.1}$$

$$w_t^i = \sum_{j \in \mathcal{N}^{-i}} A_{ij}^- x_t^j + B_{ij}^- u_t^j, \tag{7.2}$$

$$x_t^i \in X^i, \; u_t^i \in U^i \;\; \forall t, i. \tag{7.3}$$

We can eliminate the state variables in the optimization problem (7) using the dynamics (7.1). In this case we can define $\mathbf{x}^i = [w_0^{iT} \cdots w_{N-1}^{iT}\ u_0^{iT} \cdots u_{N-1}^{iT}]^T$. Then, the control problem (7) can be recast as a *separable convex quadratic program* with decoupled cost and coupled constraints in the form (**DCCC**):

$$\min_{\mathbf{x}^1,\cdots,\mathbf{x}^M} \sum_{i=1}^{M} \mathbf{x}^{iT} H_i \mathbf{x}^i + q_i^T \mathbf{x}^i \tag{8}$$
$$\text{s.t.}: \quad \mathbf{x}^i \in \mathbf{X}^i, \ \sum_{i=1}^{M} G_i \mathbf{x}^i = g,$$

where the matrices H^i are positive semidefinite, the local constraint sets $\mathbf{X}^i$ are polyhedral and the coupled constraints $\sum_{i=1}^{M} G_i \mathbf{x}^i = g$ are obtained from the coupling between the subsystems, i.e. by stacking the constraints (7.2) for all i, t. Note that the number of rows of the matrices G^i are equal to $N \sum_{i=1}^{M} p_i$.

2.3. Cooperative control problem of dynamically uncoupled systems

Cooperative control for dynamically uncoupled systems arises in a wide variety of applications like formation flying, mobile sensor networks, rendezvous problems or decentralized coordination. The cooperative control problem for dynamically uncoupled agents consists in controlling a group of independent subsystems (i.e. with decoupled dynamics), but sharing a common goal (see e.g. [9, 17, 19]).

We consider a set of M identical subsystems, having the following state-space description:

$$x_{t+1}^i = \phi(x_t^i, u_t^i), \quad y_t^i = \theta(x_t^i), \quad \forall i = 1, \cdots, M,$$

where $x_t^i \in \mathbb{R}^n$ is the state vector, $u_t^i \in \mathbb{R}^m$ is the input vector and $y_t^i \in \mathbb{R}^p$ is the output vector of subsystem i. As in the previous section we assume state and input constraints of the form (4). In the formulation of cooperative control for uncoupled systems the dynamics of subsystems are independent from each other, but they share a common goal. This calls for the minimization of a *cost function* which involves the states and inputs of each subsystem and their

neighbors as well. In this case we introduce a stage cost at time t of the form $\ell(x_t^1, \cdots, x_t^M, u_t^1, \cdots, u_t^M)$ and a final cost $\ell_f(x_N^1, \cdots, x_N^M)$.

The cooperative control problem over a finite horizon of length N, given the initial condition x^i for each subsystem i, is formulated as follows:

$$\begin{aligned} \min_{x_t^i, u_t^i} & \sum_{t=0}^{N-1} \ell(x_t^1, \cdots, x_t^M, u_t^1, \cdots, u_t^M) + \ell_f(x_N^1, \cdots, x_N^M) \\ \text{s.t. : } & x_0^i = x^i, \;\; x_{t+1}^i = \phi(x_t^i, u_t^i), \\ & x_t^i \in X^i, \; u_t^i \in U^i, \;\; \forall i, t. \end{aligned} \tag{9}$$

Now, let us denote:

$$\begin{aligned} & \mathbf{x}^i = [x_1^{iT} \cdots x_N^{iT} \;\; u_0^{iT} \cdots u_{N-1}^{iT}]^T, \\ & f(\mathbf{x}^1, \cdots, \mathbf{x}^M) = \sum_{t=0}^{N-1} \ell(x_t^1, \cdots, x_t^M, u_t^1, \cdots, u_t^M) + \ell_f(x_N^1, \cdots, x_N^M), \end{aligned}$$

and $\mathbf{X}^i$ the constraint set defined by the state and input constraints (4) and by the ith subsystem dynamics $x_{t+1}^i = \phi(x_t^i, u_t^i)$ over the prediction horizon. Using these notations, the previous cooperative control problem can be recast as an optimization problem with coupled cost and decoupled constraints (**CCDC**):

$$\textbf{(CCDC)} : \begin{cases} \min_{\mathbf{x}^1, \cdots, \mathbf{x}^M} f(\mathbf{x}^1, \cdots, \mathbf{x}^M) \\ \quad \text{s.t. : } \mathbf{x}^i \in \mathbf{X}^i. \end{cases}$$

We are interested in finding efficient parallel algorithms for solving problem (**CCDC**).

Example 2.3 We consider the formation flying for a group of satellites that are distributed along a circular orbit with independent dynamics but they have to maintain a constant distance with respect to the two nearest neighbors (see e.g. [19]). Using a discretized version of the linear Clohessy-Wiltshire equations of the ith satellite for a nominal circular trajectory [15]:

$$\begin{cases} \ddot{x}^{1,i} = 3\omega_n^2 x^{1,i} + 2\omega_n \dot{x}^{2,i} + a^{1,i} \\ \ddot{x}^{2,i} = -2\omega_n \dot{x}^{1,i} + a^{2,i} \\ \ddot{x}^{3,i} = -\omega_n^2 x^{3,i} + a^{3,i}, \end{cases}$$

where $x^{1,i}$, $x^{2,i}$, $x^{3,i}$ are the displacements in the radial, tangential and out-of-plane direction, $a^{1,i}$, $a^{2,i}$, $a^{3,i}$ represent the accelerations of the satellite i due to propulsion or external disturbances and ω_n is the angular velocity at which the orbit is covered, we obtain a discrete-time linear system for the ith satellite of the form

$$\begin{cases} x_{t+1}^i = Ax_t^i + Bu_t^i \\ y_t^i = Cx_t^i, \end{cases}$$

with $x_t^i \in \mathbb{R}^6$ and $u_t^i = [a_t^{1,i}\ a_t^{2,i}\ a_t^{3,i}]^T \in \mathbb{R}^3$ being the state, respectively the input vectors of satellite i and we consider as output $y_t^i = [x_t^{1,i}\ x_t^{2,i}\ x_t^{3,i}]^T$, the vector of absolute positions of the satellite. We also assume input constraints of the form:

$$u_{\min} \leq u_t^i \leq u_{\max} \quad \forall i, t.$$

Since the goal is to maintain a constant distance with respect to the two nearest neighbors, we choose the following stage cost at time t:

$$\ell(x_t^1, \cdots, x_t^M, u_t^1, \cdots, u_t^M) = \sum_{i=1}^{M} ||2y_t^i - y_t^{i+1} - y_t^{i-1}||_{Q_i}^2 + ||u_t^i||_{R_i}^2,$$

where Q_i, R_i are positive definite matrices. We assume the final cost $\ell_f = 0$. Despite the fact that the output y^i represents the absolute positions of the ith satellite, using the stage cost from above, the formation flying becomes a problem based on relative positions between the satellites instead of the absolute ones. In this case the cooperative control problem (9) over a finite horizon N can be recast as a *convex quadratic problem* with coupled cost and decoupled constraints in the form (**CCDC**) :

$$\begin{aligned} \min_{\mathbf{x}^1, \cdots, \mathbf{x}^M} & \begin{bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^M \end{bmatrix}^T \begin{bmatrix} \ H_{ij} \ \end{bmatrix}_{ij} \begin{bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^M \end{bmatrix} + \begin{bmatrix} q_1 \\ \vdots \\ q_M \end{bmatrix}^T \begin{bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^M \end{bmatrix} \\ \text{s.t. :} & \ \mathbf{x}^i \in \mathbf{X}^i, \end{aligned} \tag{10}$$

where the blocks of the positive semidefinite Hessian matrix $H = [H_{ij}]_{ij}$ satisfies $H_{ij} = 0$ if $|i - j| > 3$ for all i, j and the sets $\mathbf{X}^i$ are polyhedral.

Remark 2.4 (i) Note that we can eliminate the states $x_1^i, \cdots, x_N^i$ using the dynamics of the ith satellite and keeping only the inputs over the prediction horizon as decision variables, i.e. we may redefine $\mathbf{x}^i = [\ u_0^{iT} \cdots u_{N-1}^{iT}]^T$. In this case H becomes positive definite and the sets $\mathbf{X}^i$ are described only by linear inequalities.

(ii) In many applications we can move the coupling terms from the cost to the constraints by introducing auxiliary variables, i.e we can recast an optimization problem with coupled cost but decoupled constraints (**CCDC**) to one with decoupled cost but coupled constraints (**DCCC**). E.g., in our satellite formation application we can define the coupling constraints $w_t^i = y_t^{i-1} + y_t^{i+1}$ and then we can associate a local stage cost for each satellite i as $\ell^i(x_t^i, w_t^i, u_t^i) = ||2Cx_t^i - w_t^i||_{Q_i}^2 + ||u_t^i||_{R_i}^2$ but with coupled dynamics $w_t^i = C(x_t^{i-1} + x_t^{i+1})$. We can also do the other way around: we can reformulate a (**DCCC**) into a (**CCDC**) problem (e.g. by moving the coupling constraints (5.1) into the cost, see Section 2.4). Depending on applications one formulation might be preferred against the other (see also Section 2.4 below).

2.4. Cooperative control problem of dynamically coupled systems

In this section we discuss the cooperation-based optimal control problem for a group of dynamically coupled subsystems [5, 21, 39, 42, 49, 50]. For the ith subsystem we consider the following linear dynamics:

$$x_{t+1}^i = A_i x_t^i + B_i u_t^i + \sum_{j \in \mathcal{N}^{-i}} B_{ij} u_t^j, \quad \forall i = 1, \cdots, M. \tag{11}$$

Note that the dynamics described in (11) are a particular case of (6). We also assume local input constraints $u_t^i \in U^i$, where U^i are convex sets.

For each subsystem we define a local stage cost $\ell^i(x^i, u^i)$ and a terminal cost $\ell_f^i(x^i)$. The local cost for each subsystem on a finite horizon of length N will be of the following form:

$$f^i(\overline{\mathbf{x}}^i, \overline{\mathbf{u}}^i) = \sum_{i=0}^{N-1} \ell^i(x_t^i, u_t^i) + \ell_f^i(x_N^i), \tag{12}$$

where we denote with

$$\overline{\mathbf{x}}^i = [x_1^{iT} \cdots x_N^{iT}]^T, \quad \overline{\mathbf{u}}^i = [u_0^{iT} \cdots u_{N-1}^{iT}]^T. \tag{13}$$

In order to provide a cooperative behavior between subsystems we replace each local cost f^i with one that represents the systemwide impact of local control actions. One choice is to employ a strong convex combination of local subsystems' costs as the global objective function for the entire system. In these conditions, the cooperative control problem for coupled systems on a finite horizon N will have the form:

$$\min_{\overline{\mathbf{x}}^i, \overline{\mathbf{u}}^i} \sum_{i=1}^{M} \alpha_i f^i(\overline{\mathbf{x}}^i, \overline{\mathbf{u}}^i) \tag{14}$$

$$\text{s.t.}: \quad x_{t+1}^i = A_i x_t^i + B_i u_t^i + \sum_{j \in \mathcal{N}^{-i}} B_{ij} u_t^j, \quad x_0^i = x^i, \tag{14.1}$$

$$u_t^i \in U^i \quad \forall t, i, \tag{14.2}$$

where $\alpha_i > 0$ and sum to 1. Note that in this form problem (14) is a particular case of problem (**DCCC**), where the variables associated to the ith subsystem are given by $[\overline{\mathbf{x}}^{iT} \ \overline{\mathbf{u}}^{iT}]^T$. However, by eliminating the states in (14) using the global dynamic model (14.1) we obtain a coupled objective function in the local variables $\mathbf{x}^i = \overline{\mathbf{u}}^i$ (i.e. in the local control actions) and decoupled constraints, which is a particular case of (**CCDC**) problem (see also Remark 2.4(ii)).

3. Parallel and distributed optimization algorithms for solving coupled optimization problems

In this section we present several parallel and distributed algorithms for solving the optimization problems arising in applications from estimation and control discussed in Section 2 and analyze their properties and performances, in particular we define conditions for which these algorithms converge[2]. The

[2]For simplicity of the exposition, in this section we assume that all the functions are differentiable.

presented algorithms can be classified, on the one hand in "centralized" algorithms (that in general take advantage of the sparsity of the problem and solve in parallel low dimension subproblems) and distributed algorithms (that take into account explicitly information restrictions in the network and combine consensus negotiations with optimization methods to solve distributively the problem) and on the other hand in primal and dual decomposition algorithms. The first class is based on decomposing the original optimization problem, while the second consists in decomposing the corresponding dual problem.

For a given problem representation there are often many choices of distributed algorithms, each with possible different characteristics: e.g. rate of convergence, tradeoff between local computation and global communication, and quantity of message passing. Which alternative is the best depends on the specifications of the application. However, for each algorithm we will discuss in details their main characteristics in terms of performance and properties.

*3.1. Distributed gradient algorithms for optimization problems of type (**DCx**)*

In this section we study several distributed algorithms for solving separable optimization problems with decoupled cost and common decision variables in the form (**DCx**), that e.g. appear in the context of state estimation in sensor networks (see Section 2.1). We associate to the set of agents (e.g. sensors) a graph $G = (V, E)$ and then such distributed algorithms must satisfy the following constraint: the computations will be performed on all nodes in parallel, and the communication between nodes is restricted to the edges of the graph. Distributed optimization algorithms are mainly based on combining consensus negotiations (as an efficient method for information fusion) with optimization methods [14, 26, 31, 32, 52] to solve distributively problems of type (**DCx**).

First we introduce the consensus problem for a group of M agents that considers conditions under which using a certain message-passing protocol, the local variables of each agent will converge to the same value [24, 35, 51]. There exist several results related to the convergence of local variables to a common value using various information exchange protocols among agents [35, 37, 51].

One of the most used models for consensus is based on the following discrete-time iteration: to generate an estimate at iteration $k+1$, agent i forms a convex combination of its estimate $\mathbf{x}_k^i$ with the estimates received from other agents:

$$\mathbf{x}_{k+1}^i = \sum_{j=1}^{M} \gamma_k^{ij} \mathbf{x}_k^j,$$

where γ_k^{ij} represent nonnegative weights[3] satisfying $\sum_j \gamma_k^{ij} = 1$. At each iteration k the information exchange among agents can be represented by a graph (V, E_k), where $E_k = \{(i,j) : \gamma_k^{ij} > 0\}$. We can also introduce the graph (V, E_∞), where $E_\infty = \{(i,j) : (i,j) \in E_k \text{ for infinitely many } k\}$. The graphs (V, E_k) satisfy the *bounded interconnection interval property* if there exists an integer τ such that for any $(i,j) \in E_\infty$ agent j sends its information to agent i at least once every τ consecutive iterations. It has been proved in [31] that under certain assumptions on the weights γ_k^{ij} (e.g. stochasticity of the matrix $\Gamma_k = [\gamma_k^{ij}]_{ij}$, strong connectivity property of (V, E_∞) and bounded interconnection interval property), the states $\mathbf{x}_k^i$ of all agents converge to the same state x^*. Similar convergence results can be found in [24, 51].

We return now to our optimization problem of type (**DCx**). In [32] a distributed projected gradient algorithm is analyzed, which basically combines the consensus iteration presented above with a projected gradient update to generate the next estimate of the optimum. More specifically, an agent i updates its estimate by combining the estimates received from its neighbors, then taking a gradient step to minimize its objective function f^i and finally projecting on the set $\mathbf{X}$:

Algorithm dgp1

$$\mathbf{v}_k^i = \sum_{j=1}^{M} \gamma_k^{ij} \mathbf{x}_k^j, \quad \mathbf{x}_{k+1}^i = \left[\mathbf{v}_k^i - \alpha_k \nabla f^i(\mathbf{v}_k^i)\right]_{\mathbf{X}}$$

---

[3]Naturally, an agent i assigns zero weight to the estimates $\mathbf{x}^j$ for those agents j whose estimate information is not available at the update time.

where α_k is a common step size, ∇f^i denotes the gradient of the function f^i, and $[\cdot]_{\mathbf{X}}$ denotes the Euclidian projection on the set $\mathbf{X}$. The following convergence result holds for Algorithm **dgp1** :

Theorem 3.1. *[32] For the optimization problem (**DCx**) we assume that all the functions f^i are convex and have bounded gradients, the set **X** is convex and the step size satisfies $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$. Moreover, we assume that the weights γ_k^{ij} satisfy the following properties: the matrices $\Gamma_k = [\gamma_k^{ij}]_{ij}$ are doubly stochastic, the graph (V, E_∞) is connected and the bounded interconnection interval property holds. Then, the distributed projected gradient Algorithm **dgp1** converges to an optimum of problem (**DCx**).*

An interesting variant of a distributed gradient projected algorithm has been provided in [14]. Compared to the previous distributed gradient Algorithm **dgp1**, in [14] a fixed connected graph (V, E) is taken over all iterations and the information exchange among the agents is represented by a doubly stochastic matrix $\Gamma = [\gamma^{ij}]_{ij}$ such that $\gamma_{ij} > 0$ if $(i, j) \in E$. In this algorithm, first each agent implements the gradient update locally and then it runs a number μ of consensus iterations with its neighbors:

Algorithm dgp2

$$\mathbf{x}_{k+1}^i = \left[\sum_{j=1}^{M} \Gamma_{ij}^{\mu} \left(\mathbf{x}_k^j - \alpha_k \nabla f^j(\mathbf{x}_k^j) \right) \right]_{\mathbf{X}}$$

---

where Γ_{ij}^{μ} denotes the (i, j) entry of the matrix Γ^{μ}. Under similar assumptions as in Theorem 3.1, the authors in [14] proved convergence of Algorithm **dgp2** for a constant step size and for a sufficiently large μ.

In the case when the set $\mathbf{X}$ is explicitly defined through a finite set of equalities and inequalities, an algorithm based on a penalty primal-dual approach has been recently proposed in [52]. This algorithm allows the agents exchange information over networks with time-varying topologies and asymptotically agree on an optimal solution and the optimal value.

Another interesting approach for solving the optimization problem (**DCx**), but in a serial fashion, can be found in [30] where an incremental gradient method is presented. Each step of the algorithm is a gradient iteration for a single component function f^i, and there is one step per component function. Thus, an iteration can be viewed as a cycle of M subiterations, so that at $k+1$:

$$\begin{aligned} &\mathbf{x}_{k+1} = z_{M,k}, \quad z_{0,k} = \mathbf{x}_k, \\ &z_{i,k} = \left[z_{i-1,k} - \alpha_k \nabla f^i(z_{i-1,k})\right]_{\mathbf{X}} \qquad \forall i = 1, \cdots, M. \end{aligned}$$

For convex problems, using an appropriate step size α_k, the authors in [30] show that this algorithm has much better practical rate of convergence than the classical gradient method.

Remark 3.2 (i) The convexity assumptions on the functions f^i and the set $\mathbf{X}$ for convergence of the two Algorithms **dgp1** and **dgp2** are usually satisfied in many applications: see e.g. the state estimation problem for linear systems discussed in Example 2.1 which leads to the convex quadratic program (2).
(ii) One of the main challenges when solving problems of type (**DCx**) is the time-dependent communication topology, as communication links can change due to changing distances, obstacles, or disturbances. While in [14] a constant topology is assumed for Algorithm **dgp2**, the Algorithm **dgp1** and the algorithm from [52] are based on a changing topology, which makes them more suitable in practical applications. Moreover, the cyclical incremental algorithm [30] can be implemented only when each agent identifies a suitable downstream and upstream neighbor. Note the existence of a cycle is a stronger assumption than connectivity.
(iii) From simulations we have observed that the algorithms from [14, 32, 52] are very sensitive to the choice of the weights that must be tuned, since they are considered as parameters in these methods. These algorithms do not provide a mathematical way of choosing the weights from the consensus protocol, which has a very strong influence on the convergence rate of these methods. Recently in [26], a distributed algorithm has been derived for solving particular cases of

problems of type (**DCx**), where the nonnegative weights corresponding to the consensus process are interpreted as dual variables and thus they are updated using arguments from duality theory. Moreover, if the network is not densely connected (i.e. each sensor has a large number of neighbors), one can expect the performance of these algorithms from [14, 32, 52] to be worse than that of the cyclic incremental gradient [30].

M	N	nr. it. dgp1	nr. it. dgp2
10	10	5.627	586
10	20	8.447	746
20	10	10.651	1.854
20	20	14.758	2.571

Table 1: State estimation problem Example 2.1: we consider $M = 10, 20$ sensors, a linear system with 5 states and a prediction horizon $N = 10, 20$. We solve the convex quadratic program (2) with the accuracy of the solution $\epsilon = 10^{-2}$. We assume fixed weights in both algorithms such that $\gamma^{ij} = 0$ for $|i - j| > 1$ and $\mu = 10$. From simulations we observe that Algorithm **dgp2** works better than Algorithm **dgp1** in terms of the number of gradient iterations. However, Algorithm **dgp2** needs to perform for each gradient iteration also $\mu = 10$ consensus steps.

*3.2. Decomposition algorithms for solving optimization problems (**DCCC**)*

In this section we present several decomposition algorithms for solving separable optimization problems with decoupled cost but coupled constraints in the form (**DCCC**). Distributed control for complex processes with interacting subsystem dynamics usually leads to such optimization problems (see e.g. Section 2.2). We discuss two classes of decomposition principles: primal and dual. We use the terms primal and dual in their mathematical programming meaning: primal indicates that the optimization problems are solved using the original formulation and variables and dual indicates that the original problem has been rewritten using Lagrangian relaxation.

Compared to the general formulation of problem (**DCCC**), we focus in this section on decomposition methods for the particular case of separable *convex* problems with decoupled cost and coupled constraints[4]:

$$\textbf{(conv-DCCC):} \left\{ \begin{array}{l} \min_{\mathbf{x}^1,\cdots,\mathbf{x}^M} \sum_{i=1}^M f^i(\mathbf{x}^i) \\ \quad \text{s.t.}: \ \mathbf{x}^i \in \mathbf{X}^i, \ \sum_{i=1}^M G_i \mathbf{x}^i = g, \end{array} \right.$$

where we consider that for all i the coupled constraints $h^i(\mathbf{x}^j;\ j \in \mathcal{N}^i) = 0$ in problem (**DCCC**) become linear and can be written compactly as $\sum_{i=1}^M G_i \mathbf{x}^i = g$, with $G_i \in \mathbb{R}^{n_\lambda \times n_{\mathbf{x}^i}}$. For simplicity of the exposition the following assumptions hold for problem **(conv-DCCC)** (for general case of convex problems see [28, 29]):

Assumption 3.3. *Each function f^i is convex quadratic and $\boldsymbol{X}^i$ are compact convex sets. Moreover, the Slater's condition holds, i.e. there exist $\boldsymbol{x}^i \in int(\boldsymbol{X}^i)$ such that $\sum_{i=1}^M G_i \boldsymbol{x}^i = g$.*

From Example 2.2 we have seen that centralized optimal control for interconnected linear systems leads to such a separable convex quadratic formulation, e.g. (8).

We begin with primal decomposition (see e.g. [3, 7, 38, 43] and the references therein). We can decompose the original problem (**conv-DCCC**) as follows: we introduce some auxiliary variables in order to separate the coupled linear equality constraints, i.e. we introduce the new variables $\mathbf{t}^1, \cdots, \mathbf{t}^{M-1}$, and obtain M subproblems:

$$(\mathbb{P}^i): \quad \psi^i(\mathbf{t}^i) = \min_{\mathbf{x}^i} \{ f^i(\mathbf{x}^i) : \ \mathbf{x}^i \in \mathbf{X}^i, \ G_i \mathbf{x}^i = \mathbf{t}^i \}$$

for $i = 1, \cdots, M-1$ and the Mth subproblem

$$(\mathbb{P}^M): \ \ \psi^M(\mathbf{t}^1, \cdots, \mathbf{t}^{M-1}) = \min_{\mathbf{x}^M} \{ f^M(\mathbf{x}^M) : \ \mathbf{x}^M \in \mathbf{X}^M, \ \sum_{i=1}^{M-1} \mathbf{t}^i + G_M \mathbf{x}^M = g \}.$$

[4] For the nonconvex case of problem (**DCCC**) we can still obtain decomposition algorithms by combining sequential quadratic programming or sequential convex programming, in order to linearize the nonlinear coupled constraints, with decomposition methods that address the decomposable convex problems (see e.g. [27]).

The separable convex problem (**conv-DCCC**) reduces to solving the unconstrained convex *primal problem* ($\mathbb{PP}$) [43]:

$$(\mathbb{PP}): \qquad \min_{\mathbf{t}^1,\cdots,\mathbf{t}^{M-1}} \psi(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1}),$$

where $\psi(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1}) = \psi^1(\mathbf{t}^1) + \cdots + \psi^{M-1}(\mathbf{t}^{M-1}) + \psi^M(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1})$. Conditions for well-posedness of the primal problem ($\mathbb{PP}$) can be found in [43]. Let $\mathbf{x}^i(\mathbf{t}^i)$ and $\lambda^i(\mathbf{t}^i)$ be the optimal solution and the corresponding optimal Lagrange multiplier for the equality constraints $G_i\mathbf{x}^i = \mathbf{t}^i$, respectively, for subproblem $\mathbb{P}^i$ given $\mathbf{t}^i$, with $i = 1,\cdots,M-1$. Similarly, we define $\mathbf{x}^M(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1})$ and $\lambda^M(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1})$ for subproblem $\mathbb{P}^M$. Although the function ψ is potentially non smooth, assuming that Slater's condition for the convex problem (**conv-DCCC**) holds (according to Assumption 3.3), the following vector is a subgradient[5] of ψ at $(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1})$ [1, 43]:

$$\left[\lambda^M(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1}) - \lambda^1(\mathbf{t}^1)\cdots\lambda^M(\mathbf{t}^1,\cdots,\mathbf{t}^{M-1}) - \lambda^{M-1}(\mathbf{t}^{M-1})\right]^T.$$

Algorithm primal subgradient (PS)

$$\begin{aligned}
&\mathbf{x}_k^i = \mathbf{x}^i(\mathbf{t}_k^i), \quad \lambda_k^i = \lambda^i(\mathbf{t}_k^i) \ \text{ for } i = 1,\cdots,M-1\\
&\mathbf{x}_k^M = \mathbf{x}^M(\mathbf{t}_k^1,\cdots,\mathbf{t}_k^{M-1}), \quad \lambda_k^M = \lambda^M(\mathbf{t}_k^1,\cdots,\mathbf{t}_k^{M-1}),\\
&\qquad\qquad \mathbf{t}_{k+1}^i = \mathbf{t}_k^i - \alpha_k(\lambda_k^M - \lambda_k^i),
\end{aligned}$$

---

where α_k is a step size.

Remark 3.4 The step size α_k can be chosen in two ways: (i) it can vary but satisfying $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$; (ii) α_k is constant for all k.

Under Assumption 3.3 the convergence of this primal subgradient algorithm is obvious, due to the equivalence between the (**conv-DCCC**) problem and

[5] A vector $s \in \mathbb{R}^n$ is a subgradient of $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathrm{dom} f$ if for all $y \in \mathrm{dom} f$ we have $f(y) \geq f(x) + s^T(y - x)$.

the convex primal problem ($\mathbb{PP}$). When the primal problem ($\mathbb{PP}$) (called also the *master problem*) is solved using this scheme, the method has an interesting economic interpretation: at each iteration the master program allocates the resources (by choosing $\mathbf{t}_k^i$) and the nodes return the prices associated with this choice λ_k^i. The iteration continues until the prices have reached the equilibrium.

We now discuss dual decomposition [2, 7, 16, 29, 46, 48]. In dual decomposition methods we have the following economic interpretation: the master problem sets the prices for the resources to each subproblem which has to decide the amount of resources to be used depending on the price. The iteration continues until the best pricing strategy is obtained. Clearly, if the coupled constraints $\sum_i G_i \mathbf{x}^i = g$ are absent, then the problem (**conv-DCCC**) can be decoupled. Therefore it makes sense to relax these coupled constraints using duality theory. We construct the *partial augmented Lagrangian*:

$$L_\mu(\mathbf{x}, \lambda) = \sum_{i=1}^{M} f^i(\mathbf{x}^i) + \mu P_{\mathbf{X}^i}(\mathbf{x}^i) + \lambda^T (\sum_{i=1}^{M} G_i \mathbf{x}^i - g), \tag{15}$$

where $\mu > 0$ and the functions $P_{\mathbf{X}^i}$ associated to the sets $\mathbf{X}^i$ (usually called *prox functions*) must have certain properties explained below. We also define the corresponding *augmented dual function*:

$$d_\mu(\lambda) = \min_{\mathbf{x}^i \in \mathbf{X}^i} L_\mu(\mathbf{x}, \lambda), \tag{16}$$

and from the structure of L_μ we obtain that (16) decouples in M subproblems

$$\mathbf{x}^i(\mu, \lambda) = \arg \min_{\mathbf{x}^i \in \mathbf{X}^i} f^i(\mathbf{x}^i) + \mu P_{\mathbf{X}^i}(\mathbf{x}^i) + \lambda^T G_i \mathbf{x}^i.$$

We are interested in the properties of the family of augmented dual functions $\{d_\mu\}_{\mu>0}$. Note that $\lim_{\mu \to 0} d_\mu(\lambda) = d_0(\lambda)$, where $d_0(\lambda) = \min_{\mathbf{x}^i \in \mathbf{X}^i} L_0(\mathbf{x}, \lambda)$ is the standard dual function, whenever the prox functions $P_{\mathbf{X}^i}$ are chosen to be continuous on the compact sets $\mathbf{X}^i$ or are barrier functions associated to these sets (see [33]). The goal is to maximize the augmented dual function for μ sufficiently small:

$$\max_\lambda d_\mu(\lambda),$$

in order to find an approximation of the optimal Lagrange multiplier $\lambda^* = \arg\max_\lambda d_0(\lambda)$ and then to recover an approximation of the corresponding optimal primal variables $\mathbf{x}^{i*}$. We distinguish three algorithms, depending on the choice of the constant μ and of the prox functions $P_{\mathbf{X}^i}$:

(I) **dual subgradient algorithm**: $\mu = 0$ and $P_{\mathbf{X}^i} = 0$

(II) **dual fast gradient algorithm**: $\mu > 0$ and $P_{\mathbf{X}^i}$ are strongly convex functions

(III) **dual interior-point algorithm**: $\mu > 0$ and $P_{\mathbf{X}^i}$ are barrier functions for the sets $\mathbf{X}^i$.

The next theorem provides the main properties of the augmented dual function:

Theorem 3.5. *[28, 29] Under Assumption 3.3, the augmented dual function d_μ is characterized as follows:*
(I) For any $\mu \geq 0$ and convex functions $P_{\boldsymbol{X}^i}$ a subgradient of d_μ at λ is given by $\sum_i G_i \boldsymbol{x}^i(\mu, \lambda) - g$. (II) For $\mu > 0$ and strong convex functions $P_{\boldsymbol{X}^i}$ the function d_μ has a Lipschitz continuous gradient. (III) For $\mu > 0$ and barrier functions $P_{\boldsymbol{X}^i}$ the function d_μ is self-concordant.

We denote $\mathbf{x}_k^i = \mathbf{x}^i(\mu_k, \lambda_k)$. The iterations of the three algorithms are:

Algorithm dual subgradient (DS)

$$\lambda_{k+1} = \lambda_k + \alpha_k (\sum_{i=1}^{M} G_i \mathbf{x}_k^i - g)$$

Algorithm dual fast gradient (DFG)

$$\bar{\lambda}_{k+1} = \lambda_k + \frac{1}{L_{\mu_k}} (\sum_{i=1}^{M} G_i \mathbf{x}_k^i - g), \quad \lambda_{k+1} = \bar{\lambda}_{k+1} + \beta_k (\bar{\lambda}_{k+1} - \lambda_k)$$

Algorithm dual interior-point (DIP)

$$\lambda_{k+1} = \lambda_k + \alpha_k \left(\nabla^2 d_{\mu_p}(\lambda_k)\right)^{-1} \nabla d_{\mu_p}(\lambda_k) \text{ as } \mu_p \to 0,$$

---

where α_k is a step-size that can be chosen as in Remark 3.4 for algorithm (**DS**) or satisfying Armijo rule [33] for algorithm (**DIP**), L_μ is the Lipschitz constant of the gradient ∇d_μ and $\beta_k > 0$ is defined iteratively as in [33]. Moreover, in the dual interior-point algorithm (**DIP**) we have an outer iteration in p where we decrease $\mu_p \to 0$ and an inner iteration in k where we need to generate vectors close to the central path using Newton updates with $\nabla^2 d_\mu(\lambda)$ representing the Hessian of the augmented dual function d_μ at λ (see [29] for more details).

The convergence of these three algorithms (**DS**), (**DFG**) and (**DIP**) can be established under suitable assumptions on problem (**conv-DCCC**) and on the prox functions $P_{\mathbf{X}^i}$:

Theorem 3.6. *[28, 29] If Assumption 3.3 holds for the separable convex problem (**conv-DCCC**), then all three algorithms (**DS**), (**DFG**) and (**DIP**) are convergent under a suitable choice of the step-size. Moreover, the dual fast gradient algorithm (**DFG**) has complexity $\mathcal{O}(\frac{c_1}{\epsilon})$, while the dual interior-point algorithm (**DIP**) has complexity $\mathcal{O}\left(c_2 \log(\frac{c_3}{\epsilon})\right)$, where ϵ is the accuracy of the approximation of the optimum for problem (**conv-DCCC**) and c_i are some positive constants.*

We should note that in the primal subgradient algorithm we maintain feasibility of the coupled constraints in the problem (**conv-DCCC**) at each iteration while for the dual algorithms feasibility holds only at convergence of these algorithms and not at the intermediate iterations. Since for control problems the coupled constraints represent the dynamics of the networked system over the prediction horizon, when using a dual algorithm these dynamics will be satisfied only at convergence. This is a major issue when we stop at an intermediate step of a dual based algorithm.

There are also other dual decomposition methods based on the concept of augmented Lagrangians: e.g. the alternating direction method [16, 48], where a quadratic penalty term $\mu||\sum_i G_i \mathbf{x}^i - g||^2$ is added to the standard Lagrangian L_0. A computational drawback of this scheme is that the quadratic penalty term is not separable in $\mathbf{x}^i$. However, this is overcome by carrying out the

minimization problem in a Gauss-Seidel fashion, followed by a steepest ascent update of the multipliers. In other dual decomposition methods, such as partial inverse method [46] or proximal point method [6], for example a term of the form $\mu \sum_i ||\mathbf{x}^i - \mathbf{x}_k^i||^2$ is added to the Lagrangian L_0. These schemes have been shown to be very sensitive to the value of the parameter μ, with difficulties in practice to obtain the best convergence rate. Some heuristics for choosing μ can be found in the literature [6, 16, 48]. However, these heuristics have not been formally analyzed from the viewpoint of efficiency estimates for the general case (linear convergence results have been obtained e.g. only for strongly convex functions).

The new decomposition methods called here "dual fast gradient" (**DFG**) and "dual interior-point" (**DIP**) obtained by smoothing the Lagrangian are more efficient in terms of number of iterations compared to the classical primal or dual subgradient algorithm (see also Table 2). We should note however, that algorithm (**DFG**) is more appropriate than the algorithm (**DIP**) when solving problems where the number of coupling constraints is large, since for (**DIP**) we need to invert at each iteration a square matrix of dimension n_λ, where n_λ denotes the dimension of λ (or equivalently the number of rows in the matrices G_i).

It is also clear that the update rules in algorithms (**DS**) and (**DFG**) are completely distributed, according to the communication graph between subsystems. Indeed, we recall that the coupling constraints $h^i(\mathbf{x}^j;\ j \in \mathcal{N}^i) = 0$ in problem (**conv-DCCC**) are assumed to be linear, of type $G^i[\mathbf{x}^j]_{j\in\mathcal{N}^i} = g_i$, i.e. we have $[G_1 \cdots G_M] = [G^{1T} \cdots G^{MT}]^T$. Let λ^i be the Lagrange multipliers for the constraints $G^i[\mathbf{x}^j]_{j\in\mathcal{N}^i} = g_i$, and thus $\lambda = [\lambda^{1T} \cdots \lambda^{MT}]^T$. Then, the main update rules in Algorithms (**DS**) and (**DFG**) are distributed, each agent i using information only from its neighbors, e.g.:

$$\lambda_{k+1}^i = \lambda_k^i + \alpha_k \left(G^i[\mathbf{x}_k^j]_{j\in\mathcal{N}^i} - g_i \right).$$

However, for the algorithm (**DIP**), the update of the Lagrange multiplier has to be done by a central agent, i.e. in this case we have a star-shaped topology

M	N	nr. it. (**DS**)	nr. it. (**DFG**)	nr. it. (**DIP**)
10	10	5.000(0.19)	1.215(10^{-2})	78(10^{-4})
10	20	5.000(0.47)	1.873(10^{-2})	117($10^{-}4$)
10	30	5.000(0.81)	2.721(10^{-2})	165(10^{-4})

Table 2: Distributed control problem for a network of interconnected linear subsystems, Example 2.2, where $n_i = 5, m_i = 3$ and $p_i = 2$ for all i: we consider $M = 10$ subsystems and a prediction horizon $N = 10, 20$ and 30. The weighted matrices are taken $Q_i = I_5$ and $R_i = I_2$. By eliminating the states we obtain the convex quadratic program (8) with $\mathbf{x}^i = [w_0^{iT} \cdots w_{N-1}^{iT}\ u_0^{iT} \cdots u_{N-1}^{iT}]^T$, where each matrix $H_i \in \mathbb{R}^{N(m_i+p_i)\times N(m_i+p_i)}$ is positive semidefinite. In the brackets we display the accuracy ϵ. Clearly, the dual algorithms based on smoothing techniques (**DFG**) and (**DIP**) work much better than classical dual subgradient algorithm (**DS**).

for the communication among subsystems. Note that for this algorithm the sparsity of the graph will impose sparsity on the matrices G_i, which in turn will have a strong effect on the computation of the Hessian of the corresponding dual function (see [29] for more details).

*3.3. Parallel algorithms for solving optimization problems of type (**CCDC**)*

In this section we study parallel algorithms for solving optimization problems with coupled cost but decoupled constraints in the form (**CCDC**), that e.g. appear in the context of cooperative control (see Sections 2.3 and 2.4). A well known parallel algorithm in linear algebra for solving systems of linear equations is the Jacobi algorithm that can be also used in the context of optimization [2]. Applying Jacobi algorithm, we decompose our optimization problem of type (**CCDC**) into M optimization subproblems of lower dimension. In this algorithm each agent updates its variable $\mathbf{x}^i$ by solving a low dimension optimization problem where the values of the rest of variables are calculated at the previous iteration. An extension of the Jacobi algorithm is the Gauss-Seidel algorithm, where at each iteration each agent updates its variable by solving an optimization problem for which the rest of the variables are replaced with the most recent values computed. It is clear that in the Jacobi algorithm the

Algorithm Jacobi

$$\mathbf{x}_{k+1}^i = \arg\min_{\mathbf{x}^i \in \mathbf{X}^i} f(\mathbf{x}_k^1, \cdots, \mathbf{x}_k^{i-1}, \mathbf{x}^i, \mathbf{x}_k^{i+1}, \cdots, \mathbf{x}_k^M)$$

Algorithm Gauss-Seidel

$$\mathbf{x}_{k+1}^i = \arg\min_{\mathbf{x}^i \in \mathbf{X}^i} f(\mathbf{x}_{k+1}^1, \cdots, \mathbf{x}_{k+1}^{i-1}, \mathbf{x}^i, \mathbf{x}_k^{i+1}, \cdots, \mathbf{x}_k^M)$$

optimization subproblems can be solved in parallel at each iteration. The Gauss-Seidel algorithm can be also parallelized, providing that a coloring scheme can be applied (see [2] for more details).

The convergence of these two algorithms can be established under suitable contraction assumptions on the mapping $\mathbf{x} - \beta\Delta f(\mathbf{x})$ with respect to the block-maximum norm $\|\mathbf{x}\| = \max_i \|\mathbf{x}^i\|/\zeta_i$, where the ζ_i's are positive scalars and $\mathbf{x} = [\mathbf{x}^{1T} \cdots \mathbf{x}^{MT}]^T$.

Theorem 3.7. *[2]. For the optimization problem (**CCDC**) we assume that the objective function f is differentiable and suppose that the mapping $\boldsymbol{x} - \beta\Delta f(\boldsymbol{x})$ is a contraction for some positive scalar β. Then, the Jacobi and Gauss-Seidel algorithms are well defined and the sequence $\{\boldsymbol{x}_k\}_k$ converges to the minimum of (**CCDC**) linearly for both iterations.*

For the Gauss-Seidel algorithm, the assumptions for convergence given in Theorem 3.7 can be relaxed, in particular the contraction assumption can be replaced with a convexity assumption on the objective function (f needs to be differentiable and convex and, furthermore, the function f needs to be strictly convex function of $\mathbf{x}^i$ when the values of all the other components of $\mathbf{x}$ are held constant, for each i), see [2] for more details. If f is not differentiable, the Jacobi or Gauss-Seidel algorithm can fail to converge to the minimum of (**CCDC**) because it can stop at a non-optimal "corner" point at which f is non-differentiable and from which f cannot be reduced along any coordinate. The contraction assumption on the functions f for convergence of these two algorithms is usually satisfied in many applications: see e.g. the cooperative

control problem for satellite formation discussed in Example 2.3 which leads to the convex quadratic program (10) for which the Hessian satisfies the contraction assumption or the application from Section 2.4.

In [34] the optimization problem (**CCDC**) has been solved using a coordinate descent method. The iteration $k+1$ of the algorithm has the following form:

$$\mathbf{x}_{k+1}^{i_k} = \arg \min_{\mathbf{x}^{i_k} \in \mathbf{X}^{i_k}} \nabla_{i_k} f(\mathbf{x}_k)^T (\mathbf{x}^{i_k} - \mathbf{x}_k^{i_k}) + \frac{L_{i_k}}{2} \|\mathbf{x}^{i_k} - \mathbf{x}_k^{i_k}\|^2,$$
$$\mathbf{x}_{k+1}^{j} = \mathbf{x}_k^{j}, \qquad \forall j \neq i_k,$$

where i_k is chosen randomly based on a uniform distribution. Moreover, we assume componentwise Lipschitz continuity of the gradient of f with the Lipschitz constant L_i, for all $i = 1, \cdots, M$. In [34] Nesterov proves $\mathcal{O}(\frac{1}{\epsilon})$ rate of convergence in probability for the coordinate descent algorithm.

For cooperative control problems of dynamically coupled systems (see Section 2.4), which also leads to optimization problems of the form (**CCDC**), various versions of Jacobi-based algorithms have been proposed in the literature. For example in [42, 49, 50] the authors have proposed an algorithm of the following form:

$$\overline{\mathbf{x}}_k^i = \arg \min_{\mathbf{x}^i \in \mathbf{X}^i} f(\mathbf{x}_k^1, \cdots, \mathbf{x}_k^{i-1}, \mathbf{x}^i, \mathbf{x}_k^{i+1}, \cdots, \mathbf{x}_k^M),$$
$$\mathbf{x}_{k+1}^i = \alpha_i \overline{\mathbf{x}}_k^i + (1 - \alpha_i) \mathbf{x}_k^i,$$

where α_i are positive weights, summing to 1. In [42, 49, 50] the authors have shown that all the limit points of the sequence generated by the previous algorithm are optimal.

In [5] the authors have proposed a decomposition of the problem (**CCDC**) into a set of local subproblems that are solved iteratively by a network of agents. Each subproblem is obtained from (**CCDC**) discarding from the objective f the terms that do not depend on $\mathbf{x}^i$ and with the constraint set $\mathbf{X}^i$. A distributed algorithm based on the method of feasible directions has been proposed to gen-

M	N	σ	nr. it. Jacobi	nr. it. Gauss-Seidel
10	40	0.1	12.435	3.834
10	40	1	1.413	365
10	40	10	174	68

Table 3: Cooperative control problem for satellite formation Example 2.3: we consider $M = 10$ satellites and a prediction horizon $N = 40$. The weighted matrices are taken $Q_i = I_3$ and $R_i = \sigma I_3$ and the accuracy of the solution $\epsilon = 10^{-3}$. By eliminating the states we obtain the convex quadratic program (10) with $\mathbf{x}^i = [\ u_0^{iT} \cdots u_{N-1}^{iT}]^T$ and a strongly convex objective function having the convexity parameter σ. Clearly, for large σ both algorithms work better.

erate the iterations of the agents:

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \alpha_k^i(\hat{\mathbf{x}}_k^i - \mathbf{x}_k^i),$$

where the local descent direction is $d_k^i = \hat{\mathbf{x}}_k^i - \mathbf{x}_k^i$, for $\hat{\mathbf{x}}_k^i \in \mathbf{X}^i$, and the step size α_k^i satisfies the Armijo rule [33]. The local iterations require relatively low effort and arrive at a solution of (**CCDC**) at the expense of slower convergence and high communication among neighboring agents.

From the Tables 1, 2 and 3 we can observe that, in order to get an optimal solution, we need to perform a large number of iterations. Note however that in practical applications from control it is not always necessary to get an optimal solution, but we can also use a suboptimal solution that can still preserve some fundamental properties for the system such as robustness, stability, etc. Whenever a suboptimal solution is satisfactory we can stop the optimization algorithm at an intermediate iteration. Note that there exist many control strategies based on this principle of suboptimality (see e.g. [39, 45, 49]).

4. Conclusions

This paper has presented three applications from estimation and process control for networked systems that lead to coupled optimization problems with particular structure that can be exploited in decomposition algorithms. A systematic framework is then developed in the paper to explore several parallel and

distributed algorithms for solving such structured optimization problems, each with a different tradeoff among convergence speed, message passing amount, and distributed computation architecture. For each application, numerical experiments on several parallel and distributed algorithms are provided.

References

[1] D.P. Bertsekas, A. Nedic, A. Ozdaglar, Convex Analysis and Optimization, Athena Scientific, 2003.

[2] D.P. Bertsekas , J.N. Tsitsiklis, Parallel and distributed computation: Numerical methods, Prentice-Hall, 1989.

[3] S. Boyd, L. Xiao, A. Mutapcic, Notes on Decomposition Methods, Stanford University, 2003.

[4] E. Camponogara, D. Jia, B.H. Krogh, S. Talukdar, Distributed model predictive control, IEEE Control Systems Magazine, 22 (1), 44–52, 2002.

[5] E. Camponogara, L. De Oliveira, Distributed optimization for model predictive control of linear-dynamic networks, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 39 (6), 1331–1338, 2009.

[6] G. Chen, M. Teboulle, A proximal-based decomposition method for convex minimization problems, Mathematical Programming (A), 64, 81–101, 1994.

[7] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, Proceedings of the IEEE, 95 (1), 255–312, 2006.

[8] W.B. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems, IEEE Transactions on Automatic Control, 52 (7), 1249–1263, 2007.

[9] W.B. Dunbar, Distributed Receding Horizon Control of Cost Coupled Systems, Proceedings of the 46th IEEE Conference on Decision and Control, 2007.

[10] R. Ebrahimian, R. Baldick, State estimation distributed processing for power systems, IEEE Transactions on Power Systems, 15 (4), 1240–1246, 2000.

[11] M. Farina, G. Ferrari-Trecate, R. Scattolini, Moving-horizon partition-based state estimation of large-scale systems, Automatica, 46 (5), 910–918, 2010.

[12] M. Farina, G. Ferrari-Trecate, R. Scattolini, Distributed Moving Horizon Estimation for Linear Constrained Systems, IEEE Transactions on Automatic Control, 55 (11), 2462–2475, 2010.

[13] B. Johansson, A. Speranzon, M. Johansson, K. Johansson, On decentralized negotiation of optimal consensus, Automatica, 44 (4), 1175–1179, 2008.

[14] B. Johansson, T. Keviczky, M. Johansson, K.H Johansson, Subgradient methods and consensus algorithms for solving convex optimization problems, Proceedings of the 47th IEEE Conference on Decision and Control, 2008.

[15] M.H. Kaplan, Modern Spacecraft Dynamics and Controls, Wiley, 1976.

[16] S. Kontogiorgis, R. De Leone, R. Meyer, Alternating direction splittings for block angular parallel optimization, Journal of Optimization Theory and Applications, 90 (1), 1–29, 1996.

[17] T. Keviczky, F. Borrelli, G.J. Balas, Decentralized receding horizon control for large scale dynamically decoupled systems, Automatica, 42 (12), 2105–2115, 2006.

[18] L.S. Lasdon, Optimization Theory for Large Systems, Macmillan Series for Operations Research, 1970.

[19] P. Massioni, T. Keviczky, M. Verhaegen, New Approaches to Distributed Control of Satellite Formation Flying, Proceedings of the 3rd International Symposium on Formation Flying, Missions and Technologies, 2008.

[20] D. Madjidian, A. Rantzer, A. El-Shaer, A Wind Field Model for Distributed Control of Wind Farms, Technical Report University Lund, Sweden, 2010.

[21] J.M. Maestre, D.M. de la Pea, E.F. Camacho, Distributed MPC based on a cooperative game, Optimal Control Applications and Methods, 2010.

[22] M.D. Mesarovic, D. Macko, Y. Takahara, Theory of Hierarchical, Multilevel, Systems, Academic Press, 1970.

[23] M.S. Mahmoud, Multilevel systems control and applications: A survey, IEEE Transactions on Systems, Man, and Cybernetics, 7 (3), 125–143, 1977.

[24] L. Moreau, Stability of muliagent systems with time-dependent communication links, IEEE Transactions on Automatic Control, 50 (2), 169–182, 2005.

[25] R. Negenborn, B. De Schutter, J. Hellendoorn, Multi-Agent Model Predictive Control For Transportation Networks: Serial versus Parallel Schemes, Engineering Applications of Artificial Intelligence, 21 (3), 353–366, 2008.

[26] I. Necoara, I. Dumitrache, J.A.K. Suykens, Fast primal-dual distributed projected linear iterations for constrained consensus, Proceedings of the 49th IEEE Conference on Decision and Control, 2010.

[27] I. Necoara, C. Savorgnan, Q. Tran-Dinh, J. Suykens, M. Diehl, Distributed Nonlinear Optimal Control using Sequential Convex Programming and Smoothing Techniques, Proceedings of the 48th IEEE Conference on Decision and Control, 2009.

[28] I. Necoara, J.A.K. Suykens, Application of a smoothing technique to decomposition in convex optimization, IEEE Transactions on Automatic Control, 53 (11), 2674–2679, 2008.

[29] I. Necoara, J.A.K. Suykens, An interior-point Lagrangian decomposition method for separable convex optimization, Journal of Optimization Theory and Applications, 143 (3), 567–588, 2009.

[30] A. Nedic, D.P. Bertsekas, Incremental Subgradient Methods for Nondifferentiable Optimization, SIAM Journal on Optimization, 12 (1), 109–138, 2001

[31] A. Nedic, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, IEEE Transactions on Automatic Control, 54 (1), 48–61, 2009.

[32] A. Nedic, A. Ozdaglar, P. Parrilo, Constrained Consensus and Optimization in Multi-Agent Networks, IEEE Transactions on Automatic Control, 55 (4), 922–938, 2009.

[33] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Kluwer, 2004.

[34] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, in CORE Discussion Papers, no. 20100022010, 2010.

[35] R. Olfati-Saber, J. Fax, R. Murray, Consensus and Cooperation in Networked Multi-agent Systems, Proceedings of the IEEE, 95 (1), 215–233, 2007.

[36] L.B. Oliveira, E. Camponogara, Multi-agent model predictive control of signaling split in urban traffic networks, Transportation Research Part C: Emerging Technologies, 18 (1), 120–139, 2010.

[37] A. Olshevsky, J. Tsitsiklis, Convergence rates in distributed consensus and averaging, Proceedings of the 45th IEEE Conference on Decision and Control, 2006.

[38] D.P. Palomar, M. Chiang, A Tutorial on Decomposition Methods for Network Utility Maximization, IEEE Journal on Selected Areas in Communications, 24 (8), 1439–1451, 2006.

[39] G. Pannocchia, S.J. Wright, B.T. Stewart, J.B. Rawlings, Efficient cooperative distributed MPC using partial enumeration, Proceedings of ADCHEM 2009 (International Symposium on Advanced Control of Chemical Processes), 2009.

[40] C.V. Rao, J.B. Rawlings, D.Q. Mayne, Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations, IEEE Transactions on Automatic Control, 48 (2), 246–258, 2003.

[41] C.V. Rao, Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems, Ph.D. dissertaion, University Wisconsin-Madison, 2000.

[42] J.B. Rawlings, B.T. Stewart, Coordinating multiple optimization-based controllers: New opportunities and challenges, Journal of Process Control, 18, 839–845, 2008.

[43] G.J. Silverman, Primal decomposition of mathematical programs by resource allocation: I basic theory and a direction-finding procedure, Operations Research 20 (1) , 58–74, 1972.

[44] R. Scattolini, Architectures for distributed and hierarchical model predictive control - a review, Journal of Process Control, 19, 723–731, 2009.

[45] P.O.M. Scokaert, D.Q. Mayne, J.B. Rawlings, Suboptimal model predictive control (feasibility implies stability), IEEE Transactions on Automatic Control, 44 (3), 648–654, 1999.

[46] J.E. Spingarn, Applications of the method of partial inverses to convex programming: decomposition, Mathematical Programming (A), 32, 199–223, 1985.

[47] H. Tamura, Decomposition techniques in large scale systems, Systems and Control, 17 (6), 1973.

[48] P. Tseng, Applications of a splitting algorithm to decomposition in convex programming and variational inequalities, SIAM Journal of Control and Optimization, 29 (1), 119–138, 1991.

[49] A.N. Venkat, J.B. Rawlings, S.J. Wright, Distributed model predictive control of large-scale systems, Assessment and Future Directions of Nonlinear Model Predictive Control, 2005.

[50] A.N. Venkat, Distributed Model Predictive Control: Theory and Applications, Ph.D. dissertaion, University Wisconsin-Madison, 2006.

[51] F. Xiao, L. Wang, Asynchronous consensus in continuous-time multiagent systems with switching topology and time-varying delays, IEEE Transactions on Automatic Control, 53 (8), 1804-1816, 2008.

[52] M. Zhu, S. Martinez, On distributed optimization under inequality and equality constraints via penalty primal-dual methods, Proceedings of American Control Conference, 2434–2439, 2010.